ARCHER Hardware

Overview and Introduction

Slides contributed by Cray and EPCC

Reusing this material

archer

epcc

THE UNIVERSITY OF EDINBURGH

Nodes: The building blocks

The Cray XC30 is a Massively Parallel Processor (MPP) supercomputer design. It a distributed memory system built from thousands of individual shared-memory nodes.

There are two basic types of nodes in any Cray XC30:

- Compute nodes
  - These only do user computation, also referred to as the “back-end”
- Service/Login nodes
  - e.g. the ARCHER “front-end”: **`login.archer.ac.uk`**
  - These provide all the additional services required for the system to function, and are given additional names depending on their individual task:
    - Login nodes – allow users to log in and perform interactive tasks
    - PBS Mom nodes – run and managing PBS batch scripts
    - Service Database node (SDB) – holds system configuration information
    - LNET Routers - connect to the external filesystem.
- There are usually many more compute than service nodes

Differences between Nodes

Service/Login Nodes

- The node you access when you first log in to the system.
- Run a full version of the CLE Linux OS (all libraries and tools available)
- Used for editing files, compiling code, submitting jobs to the batch queue and other interactive tasks.
- Shared resources that may be used concurrently by multiple users.
- There may be many service nodes in any Cray XC30 and can be used for various system services (login nodes, IO routers, daemon servers).

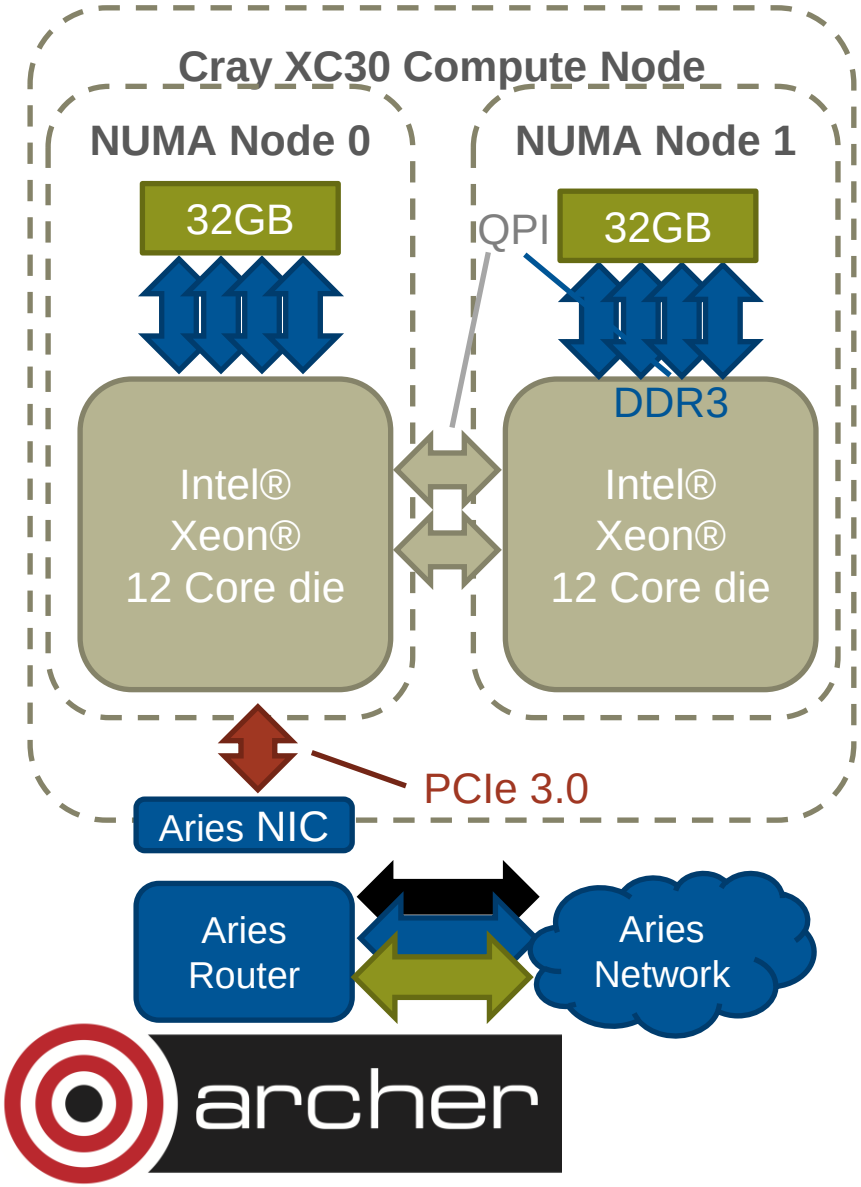
Compute nodes

- These are the nodes on which production jobs are executed
- They run Compute Node Linux, a version of the Linux OS optimised for running batch workloads
- Can only be accessed by submitting jobs to a batch management system (PBS Pro on ARCHER)
- Exclusive resources, allocated (by PBS) to a single user at a time.
- Many more compute nodes in any Cray XC30 than login / service nodes.

ARCHER Layout

Compute node architecture and topology

Cray XC30 Intel® Xeon® Compute Node



The XC30 Compute node features:

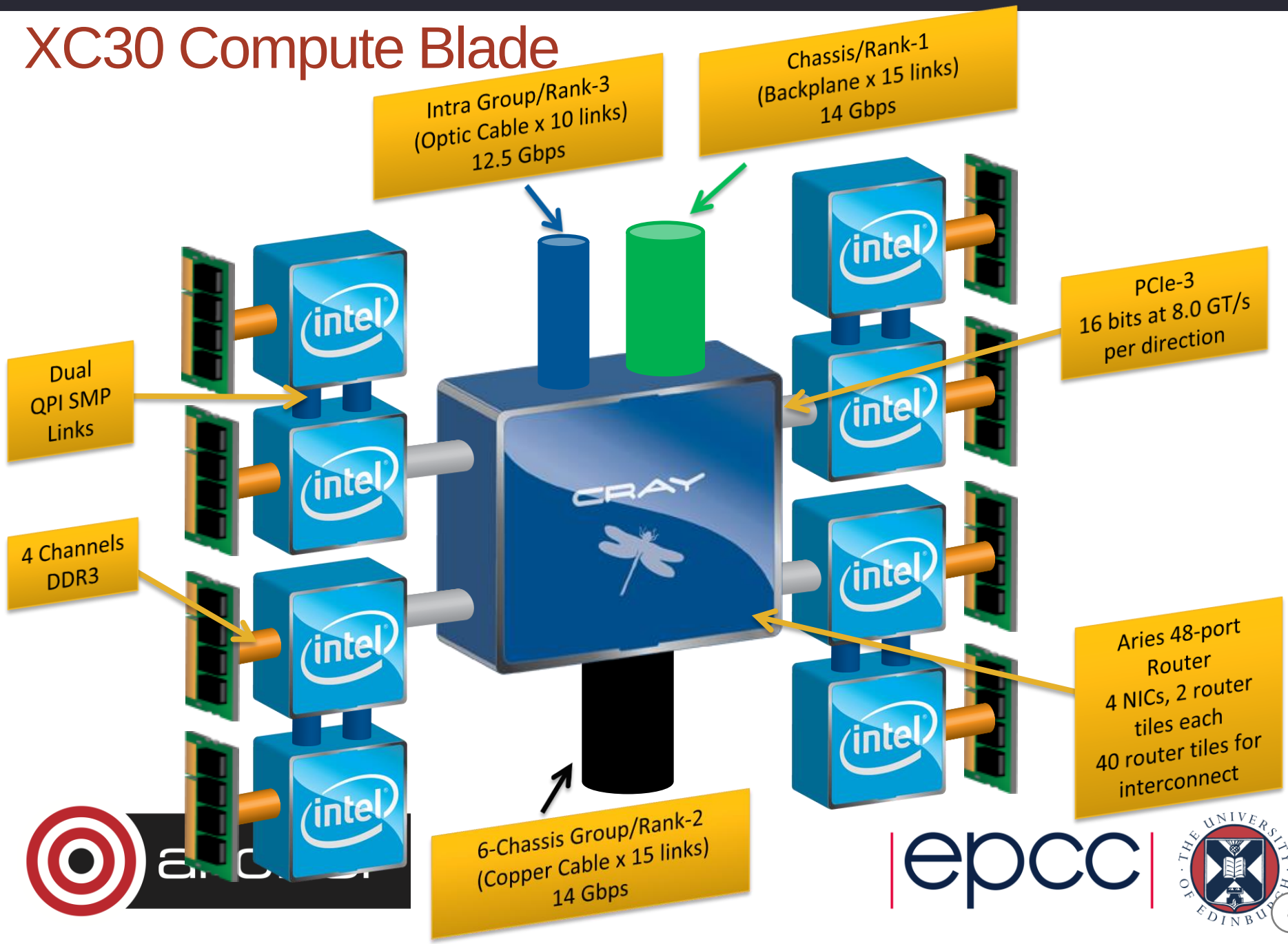
- 2 x Intel® Xeon® Sockets/die
  - 12 core Ivy Bridge
  - QPI interconnect
  - Forms 2 NUMA regions
- 8 x 1833MHz DDR3
  - 8 GB per Channel
  - 64/128 GB total
- 1 x Aries NIC
  - Connects to shared Aries router and wider network
  - PCI-e 3.0

epcc | THE UNIVERSITY OF EDINBURGH

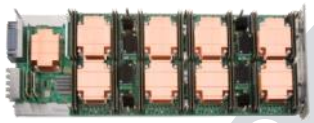
Terminology

- A *node* corresponds to a single Linux OS
  - on ARCHER, two sockets each with a 12-core CPU
  - all cores on a node see the same shared memory space
  - ARCHER comprises many 24-core shared-memory systems
  - i.e. maximum extent of an OpenMP shared-memory program
- Packaging into compute nodes is visible to the user
  - minimum quantum of resources allocation is a node
  - user given exclusive access to all cores on a node
  - ARCHER resources requested in multiples of nodes
- Higher levels (blade/chassis/group) not explicitly visible
  - but may have performance impacts in practice

XC30 Compute Blade

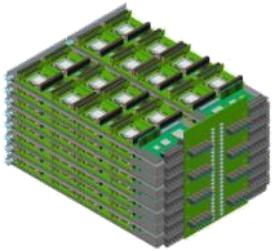


ARCHER System Building Blocks



Compute Blade

4 Compute Nodes



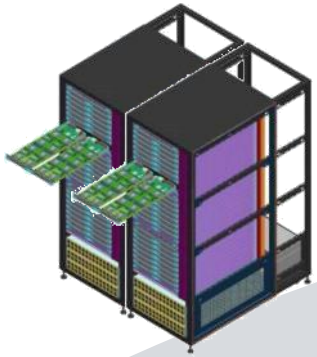
Chassis

Rank 1 Network

16 Compute Blades

No Cables

64 Compute Nodes



Group

Rank 2 Network

Passive Electrical Network

2 Cabinets

6 Chassis

384 Compute Nodes

System

Rank 3 Network

Active Optical Network

12 Groups

4920 Compute Nodes

archer

epcc

THE UNIVERSITY OF EDINBURGH

Cray XC30 Dragonfly Topology + Aries

EPSRC

NERC SCIENCE OF THE ENVIRONMENT

archer

CRAY THE SUPERCOMPUTER COMPANY

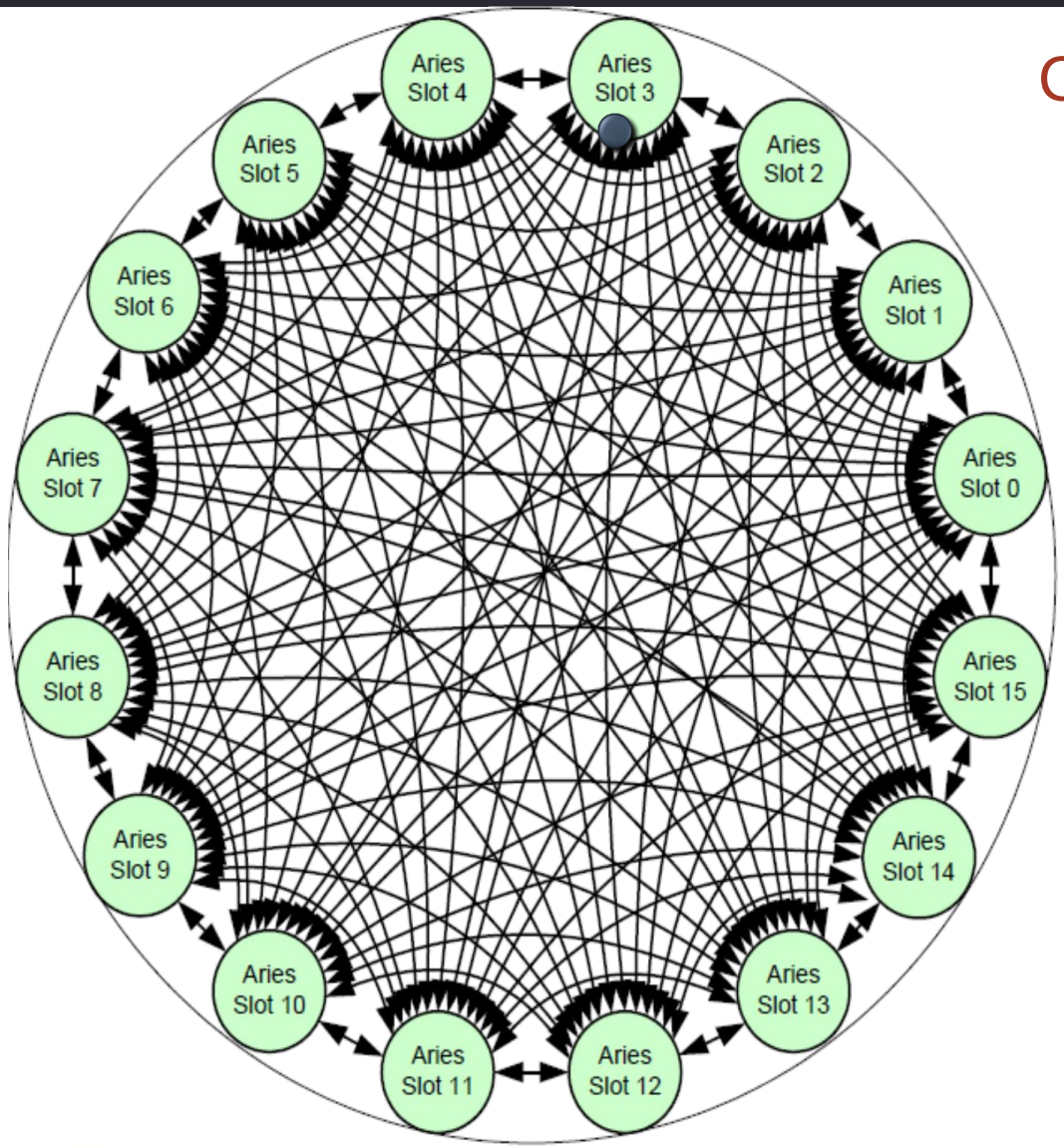
epcc

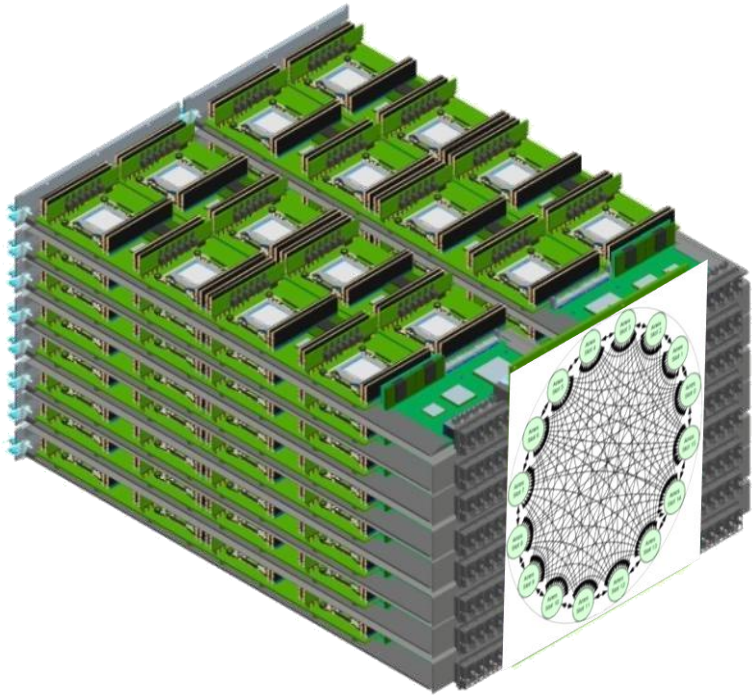
THE UNIVERSITY OF EDINBURGH

Cray Aries Features

- Scalability to > 500,000 X86 Cores
  - Cray users run large jobs – 20-50% of system size is common
  - Many examples of 50K-250K MPI tasks per job
  - Optimized collectives MPI_Allreduce in particular
- Optimized short transfer mechanism (FMA)
  - Provides global access to memory, used by MPI and PGAS (OpenSHMEM, UPC, Fortran coarrays, ...)
  - High issue rate for small transfers: 8-64 byte put/get and atomic memory operations in particular
- HPC optimized network
  - Small packet size 64-bytes
  - Router bandwidth >> injection bandwidth
  - Adaptive Routing & Dragonfly topology
- Connectionless design
  - Doesn't depend on a connection cache for performance
  - Limits the memory required per node
- Fault tolerant design
  - Link level retry on error
  - Adaptive routing around failed links
  - Network reconfigures automatically (and quickly) if a component fails
  - End to end CRC check with automatic software retry in MPI

Cray XC30 Rank1 Network





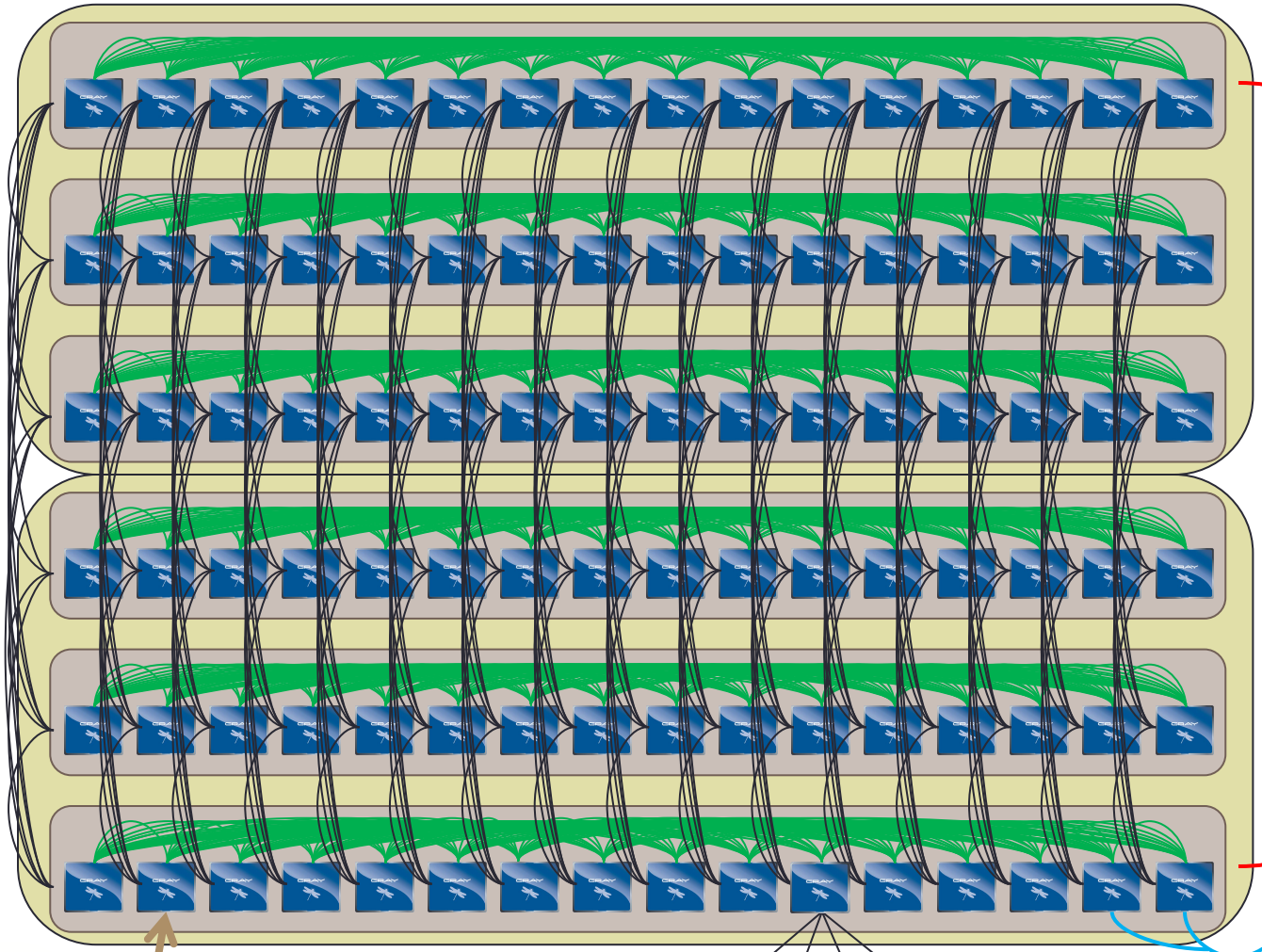
- **Chassis with 16 compute blades**
- **128 Sockets**
- **Inter-Aries communication over backplane**
- **Per-Packet adaptive Routing**

archer

epcc

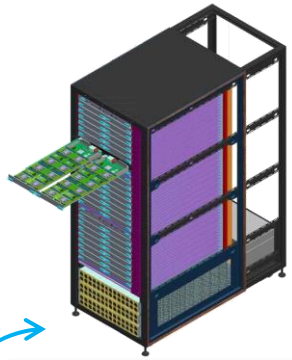
THE UNIVERSITY OF EDINBURGH

Cray XC30 Rank-2 Copper Network



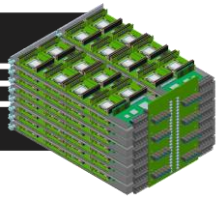
2 Cabinet Group
768 Sockets

6 backplanes connected with copper cables in a 2-cabinet group:



Active optical cables interconnect groups

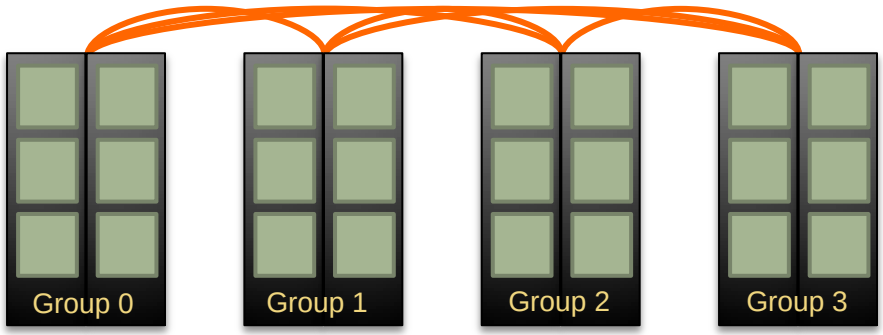
16 Aries connected by backplane



4 nodes connect to a single Aries

Cray XC30 Network Overview – Rank-3 Network

- An all-to-all pattern is wired between the groups using optical cables (blue network)
- Up to 240 ports are available per 2-cabinet group
- The global bandwidth can be tuned by varying the number of optical cables in the group-to-group connections



Example: A 4-group system is interconnected with 6 optical "bundles". The "bundles" can be configured between 20 and 80 cables wide

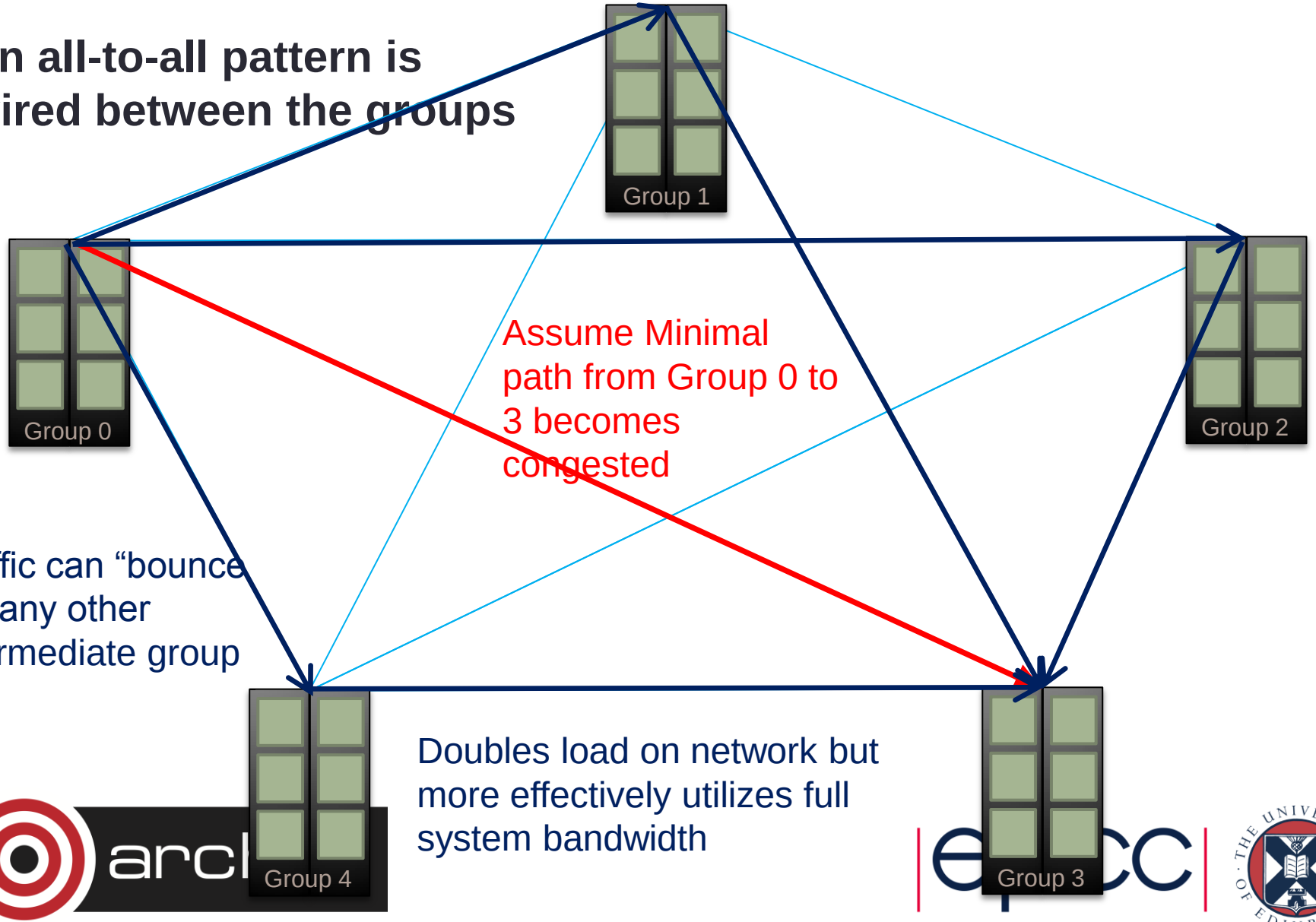
archer

epcc

THE UNIVERSITY OF EDINBURGH

Adaptive Routing over optical network

- **An all-to-all pattern is wired between the groups**



Filesystems

- /home – NFS, not accessible on compute nodes
  - For source code and critical files
  - Backed up
  - > 200 TB total
- /work – Lustre, accessible on all nodes
  - High-performance parallel filesystem
  - Not backed-up
  - > 4PB total
- RDF – GPFS, not accessible on compute nodes
  - Research Data Facility
  - Long term data storage

Filesystems

- No /tmp on backend nodes
- Users assigned to projects: filesystems based around projects:
  - /home/projectcode/projectcode/username
  - /work/projectcode/projectcode/username
- Group permissions also done per project
- Sharing data
  - Within projects
    - /work/projectcode/projectcode/shared
    - /home/projectcode/projectcode/shared
  - Between projects
    - /work/projectcode/shared
    - /home/projectcode/shared

Summary of ARCHER

- Each node contains 24 Intel IvyBridge cores
- 4920 Compute Nodes connected by Aries network
  - 64 GB per node; 1/12th of the nodes (one group) have 128 GB
- Total of 118,080 cores
  - over 300 TB memory
- Peak performance of 2.55 PF

ARCHER Software

Brief Overview

Cray's Supported Programming Environment

Programming Languages

- Fortran
- C
- C++
- Python

Programming models

- Distributed Memory (Cray MPT)
  - MPI
  - SHMEM
- Shared Memory
  - OpenMP 3.0
  - OpenACC
- PGAS & Global View
  - UPC (CCE)
  - CAF (CCE)
  - Chapel

Compilers

- Cray Compiling Environment (CCE)
- GNU
- 3rd Party Compilers
  - Intel Composer

Tools

Environment setup
- Modules

Debuggers
- Allinea (DDT)
- lgdb

Debugging Support Tools
- Abnormal Termination Processing
- STAT

Performance Analysis
- CrayPat
- Cray Apprentice²

Scoping Analysis
- Reveal

Optimized Scientific Libraries

- LAPACK
- ScaLAPACK
- BLAS (libgoto)
- Iterative Refinement Toolkit
- Cray Adaptive FFTs (CRAFFT)
- FFTW
- Cray PETSc (with CASK)
- Cray Trilinos (with CASK)

I/O Libraries

- NetCDF
- HDF5

Cray developed
Licensed ISV SW
3rd party packaging
Cray added value to 3rd party

archer

epcc

THE UNIVERSITY OF EDINBURGH

Cray MPI

- Cray MPI
  - Implementation based on MPICH2 from ANL
  - Includes many improved algorithms and tweaks for Cray hardware
    - Improved algorithms for many collectives
    - Asynchronous progress engine allows overlap of computation and comms
    - Customizable collective buffering when using MPI-IO
    - Optimized Remote Memory Access (one-sided) fully supported including passive RMA
  - Full MPI-2 support with the exception of
    - Dynamic process management (MPI_Comm_spawn)
  - MPI-3 support coming soon

Cray Performance Analysis Tools (PAT)

- From performance measurement to performance analysis
- Assist the user with application performance analysis and optimization
  - Help user identify important and meaningful information from potentially massive data sets
  - Help user identify problem areas instead of just reporting data
  - Bring optimization knowledge to a wider set of users
- Focus on ease of use and intuitive user interfaces
  - Automatic program instrumentation
  - Automatic analysis
- Target scalability issues in all areas of tool development

Debuggers on Cray Systems

- Systems with hundreds of thousands of threads of execution need a new debugging paradigm
  - Innovative techniques for productivity and scalability
    - Scalable Solutions based on MRNet from University of Wisconsin
    - STAT - Stack Trace Analysis Tool
      - Scalable generation of a single, merged, stack backtrace tree
      - running at 216K back-end processes
    - ATP - Abnormal Termination Processing
      - Scalable analysis of a sick application, delivering a STAT tree and a minimal, comprehensive, core file set.
- Support for traditional debugging mechanism
  - Allinea DDT 4.0.1
  - gdb

User administration

- SAFE website used for user administration
  - https://www.archer.ac.uk/safe
- Apply for accounts
- Manage project resources
- Report on usage
- View queries
- Etc.…